ISSN 1000-0054 清华大学学报 (自然科学版) 2005年 第 45卷 第 7期 29/38
CN 11-2223/N J Tsinghua Univ (Sci & Tech), 2005, Vol. 45, No. 7 973-976

嵌入式 CPU软硬件协同开发中的操作系统设计

董 渊, 王生原, 陈 嘉, 田金兰, 张素琴

(清华大学 计算机科学与技术系, 北京 100084)

摘 要: 为了满足高性能嵌入式 CPU软硬件协同开发的需要,提出一个嵌入式 Linux操作系统设计方案,在真正的硬件完成之前利用虚拟原型系统进行软硬件集成测试。该方案基于开放源代码软件,采用精简配置的 Linux Kernel,以 uClibc和 Busybox为主构成根文件系统,特别选择加入必要的基准测试程序。该系统成功应用于清华大学 THUMP系列 CPU开发,保证了验证的完备性,提高了验证效率,为 CPU的性能优化提供了有力的支持。实验结果表明:该方案满足了验证目的和虚拟环境对操作系统设计提出的严格要求,同时为目标 CPU未来运行系统提供了基础。

关键词: 操作系统; 嵌入式; 软硬件协同开发; 系统设计

中图分类号: TP 316 文献标识码: A

文章编号: 1000-0054(2005) 07-0973-04

Embedded Linux system design for hardware/software co-development

DONG Yuan, WANG Shengyuan, CHEN Jia,
TIAN Jinlan, ZHANG Suqin

(**Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China**)

Abstract The development of high performance embedded CPU requires co-development of the hardware, the development kits and the operating system. This paper describes a design scheme for an embedded Linux system that provides co-design of all three parts and verification in a virtual hardware environment before the physical hardware is available. Based on open source softwore the scheme uses a simplified Linux kernel with an embedded root file system including uClibc, Busybox and some benchmark programs from SPEC2000. This system has been successfully applied to the hardware/software co-development of Tsinghua University's MicroProcessor (THUMP). The system improves the verification efficiency and optimizes the performance of the embedded CPU. Test results show that the system can correctly and efficiently perform co-verification on the cycle-accurate simulator, meeting the strict design constraints of simulatnion development enviroment and successful running of the target CPU.

Key words operation system; embedded; hardware/software co-development; system design

现代生产生活中广泛使用的嵌入系统大多采用软硬件协同设计方法开发,以加快产品上市时间,提高质量并满足更为严格的设计约束[1]。协同设计中必须考虑系统建模、软硬件之间的功能划分、协同综合、协同模拟和协同验证等问题。其中协同验证利用虚拟的硬件原型系统(软件模拟器或者硬件仿真器),在真正的硬件实现之前进行系统集成测试,是整个方法中最重要的步骤之一。

高性能嵌入式微处理器需要自己的操作系统,在这些 CPU的协同开发过程中必须同时考虑操作系统的设计,验证阶段也采用该系统。

通常的嵌入式操作系统开发使用经过验证的硬件环境和开发环境。通常嵌入式系统协同设计开发中, CPU和相应的开发环境也是已有的并且经过验证,软件系统和硬件同步设计和开发。嵌入 CPU软硬件协同开发不同于上述两种情况, CPU本身、相应的开发工具以及操作系统同步开发设计,三者都未经验证。

本文所讨论的嵌入式操作系统开发就是处于嵌入式 CPU设计这个特定情形下,硬件环境、开发环境以及操作系统都统一纳入到协同设计中,功能和接口划分清楚之后同步并行开发,在各种模拟器环境中运行操作系统进行协同验证。因此,这样的操作系统是在软硬件协同开发过程中,使用不完善的工具开发,运行于不完善的模拟器平台之上,在开发阶段作为开发工具和硬件平台的一个测试向量,验证开发工具以及 CPU的设计和实现,特别 CPU中关于软硬件接口、系统初始化以及例外处理的部分,在硬件完成之后,经过扩展作为开发板的操作系统。

收稿日期: 2004-06-10

基金项目: 国家自然科学基金资助项目 (60083004);
国家"八六三"高技术项目 (2002AA1Z030)

作者简介: 董渊 (1973-), 男 (汉), 内蒙古, 讲师。

本文以清华大学高性能、低功耗的嵌入式微处理器 THUMP(Tsinghua University microprocessor)协同开发为背景,提出一种 TUES(Tsinghua University embedded system) Linux操作系统设计方案,详细介绍该方案的关键方法和技术问题。

1 系统设计目标

在软硬件协同开发中对操作系统的要求是能够达到硬件环境、开发环境以及操作系统互相验证的目的,验证目的和虚拟运行环境决定了系统的设计目标。

要达到验证目的,必须对作为测试向量的操作系统的代码以及运行情况有深入的把握,这就要求操作系统要尽可能小,便于把握和跟踪,同时功能要尽可能完备,以保证测试尽可能全面。

协同验证采用虚拟环境,对软件有非常严格的空间和时间要求。许多传统软硬件协同开发方法和工具都基于高层描述语言(HDL),这些方法协同验证通常使用硬件仿真器或 RTL(register transport level)模拟器[2]。最近的研究工作提出基于 C/C++ 的方法[3]用于软硬件协同设计,以便能够更早地开始软硬件的协同验证和性能分析,这类方法用于协同验证的模拟器可以是周期精确的 C模拟器、RTL 精确的 C模拟器、周期和相位精确模拟器等。通常情况下这些高级语言编写的模拟器的执行速度在 1 到 10k cycles/s的量级[4],而 RTL模拟器的执行速度更低。可见,时间效率是最核心的问题。

因此,用于协同开发的操作系统的设计目标可以归纳为"具体而微,时间第一",在保证系统具备基本功能的前提下尽可能地小,时间效率的优先级最高,必要情况下可以牺牲空间来换取时间效率,以便争取更多的时间进行硬件设计优化。

2 系统开发环境

THUMP和 MIPS主流产品 4KC[5]兼容,使用和 4KC不同的先进流水线结构,采用 MIPS32指令集。开发环境以 GNU工具为基础开发。开发中,首先实现周期精确的 C模拟器,完整模拟所有指令的执行情况,同时实现了单步执行、数据和指令断点、状态保存与恢复等非常强大的调试功能,以实现软硬件协同验证的需要。

整个系统开发、调试均采用交叉方式,宿主机为 PC机上的 RedHat Linux,交叉开发环境基于开放源代码的 GCC(GNU's compiler collection),目标机是运行于 PC Windows系统上周期精确的 C模拟器。图 1描述了操作系统的交叉开发和协同验证环境。

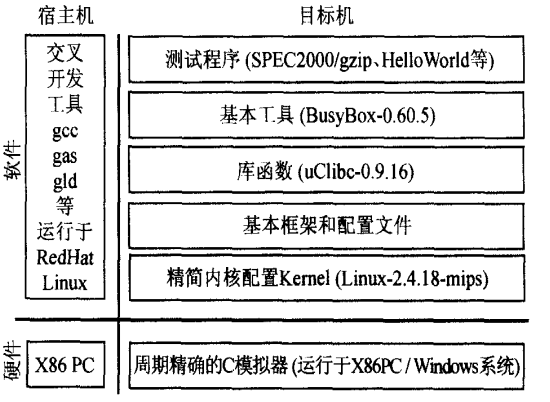

图 1 嵌入式 CPU协同开发中的操作系统开发环境

3 操作系统设计方案

本文方案选择基于 Linux自行开发嵌入式操作系统系统。Linux[6]是使用最广泛的嵌入式操作系统之一,属于开放源代码软件,可以根据实际需要裁减和修改,应用程序重用机会大、开发风险小。

一个完整的 Linux操作系统除内核 Linux Kernel之外,必须包括根文件系统,根文件系统通常包含基本框架(目录结构,启动配置文件)、库函数、init、shell、文件管理、用户管理、进程管理等多个必要的工具包,以及针对特定用户的应用程序。本文设计工作在上述几个方面根据协同开发的特定情况进行处理,确定设计方案。精简的 Linux Kernel支持尽可能少的设备,由 uClibc[7]和 Busybox[8]为主构成根文件系统,只保留少量的必要指令,同时加入部分基准测试程序,文件系统装在内嵌的 RAMDISK(采用内存 RAM模拟的硬盘 DISK)中,它们之间的关系参见图 1。

3.1 Linux Kernel配置

采用 MIPS公司发布的内核 linux-2.4.18-mips-01.00为基础版本,深入研究 Linux系统的启动过程、例外处理流程、时钟与中断、输入/输出(串口与键盘)等方面的内容。

采用尽可能精简的内核配置方案,去掉所有不必要的核心功能,减少内核开销。确定的内核配置方案为:支持串口/键盘/VGA/Timer等设备;支持

RAM /PROC/EXT2文件系统,支持可执行文件格式为 ELF; Kernel本身的格式可以是 elf32, ecoff, srec, 小模式; 支持 RAMDISK /INITRD机制; 不支持核心模块动态加载。

针对协同开发的要求和 THUMP的特点,对底层汇编代码进行必要的移植。移植工作着重处理的代码集中在例外处理及返回代码、含指令互锁的汇编代码、初始化过程中外围设备相关汇编代码、软件中其他外围设备相关汇编代码、基本函数库等方面,涉及到 17个目录下的 20多个 C和汇编文件。

3.2 基本框架和 C lib

根文件系统的基本框架包括必要的目录结构和位于 /etc目录下的启动配置文件。配置文件控制着系统的启动流程,也是不同 Linux发行版本之间的主要差别所在。本文的设计方案对配置文件进行最大可能的简化,使用 4个简单脚本文件就完成了所有的启动配置,它们是 /etc/inittab, /etc/init.d/rcS, /etc/default/rcS和 /etc/init.d/mountall.sh。

Linux Kernel引导完成由 1号进程读取 /etc/inittab文件,按照该文件来执行相应的动作。本方案的 inittab只处理进入 shell之前的初始化工作和启动 shell的工作。前者启动 /etc/init.d/rcS文件中指定的各种服务,本方案关键任务是完成文件系统加载,后者在回车确认之后开始交互执行命令。

采用针对嵌入式应用的 C函数库 uClibc, 对底层汇编代码进行必要的移植和优化。

3.3 系统工具

通常商用的 Linux操作系统使用 SysVinit、shell、fileutils、sh-utils、util-linux、vim等多个必要的工具包,需要占用大量存储空间。本方案采用 Busybox, 将通常使用到的 UNIX工具整合到一个可执行文件中,作为所有工具的最小替代品,为小规模系统提供了一个相当完整的标准环境。

使用 uClibc函数库来编译连接 Busybox, 对底层汇编代码进行必要的移植,使用尽可能小的配置方案,只保留 shell、more、vi等 20多个基本工具。

3.4 测试程序

选择测试程序加入到根文件系统是本文方案的一个主要特点,也是协同开发的关键需求之一。选择 HelloWorld测试输出,选择含有递归算法的八皇后程序以便进行函数递归调用方面的测试,选择基准测试程序 SPEC2000中 gzip程序进行大规模数值计算测试。

方案内容和空间占用情况如表 1所示,包括上述所有基本框架、库函数、busybox工具集以及测试程序构成完整的根文件系统,装载根文件系统的 RAMDISK未经压缩时 1.6 MB, 压缩之后 340 kB, 而用于类似开发板、具有类似功能的商用嵌入式系统由于采用标准的 glibc和 SysVinit、shell、fileutils、sh-utils等工具包,其压缩格式的根文件系统高达 60 MB。

表 1 嵌入式操作系统 Tues Linux设计方案

内容	方案	基准版本	Size/kB	备注
Linux Kernel	linux-2.4.18	mips-01.00	2900	代码移植,优化。含 RAMDISK
根文件系统	RAMDISK	linux内嵌	1600	含下列所有内容,未压缩
基本框架	Tues	1.1	80	含优化的启动配置文件
libc	uClibc	0.9.16	780	底层代码移植优化
工具	BusyBox	0.60.3	400	底层代码移植优化
应用程序	测试程序	1.1	300	SPEC2000(gzip)、HelloWorld、Queen等

4 协同验证过程

协同验证工作基于周期精确的 C模拟器开展,对整个系统的执行过程进行反汇编级的跟踪调试,形成一系列测试文档,发现并完整记录模拟器、编译工具以及操作系统设计和编码中的大量 BUG, 为系统的修改完善提供了重要依据。

基于 C模拟器的协同验证工作大致经历了 4个阶段,见图 2。第 1阶段, Kernel系统初始化,完成 CPU自身初始化、内存初始化、例外处理程序注册、系统设备的初始化等。第 2阶段,根文件系统加载和第一个用户进程启动,实现内核态到用户态的转换。第 3阶段,所有可以人工引发例外的测试。第 4阶段,系统运行测试。系统正常运行情况下,以交互方式执行包含 ash和文件/目录操作、文件编辑等

多种命令,以及八皇后问题、HelloWorld等应用程序和SPEC2000中的部分基准测试程序。

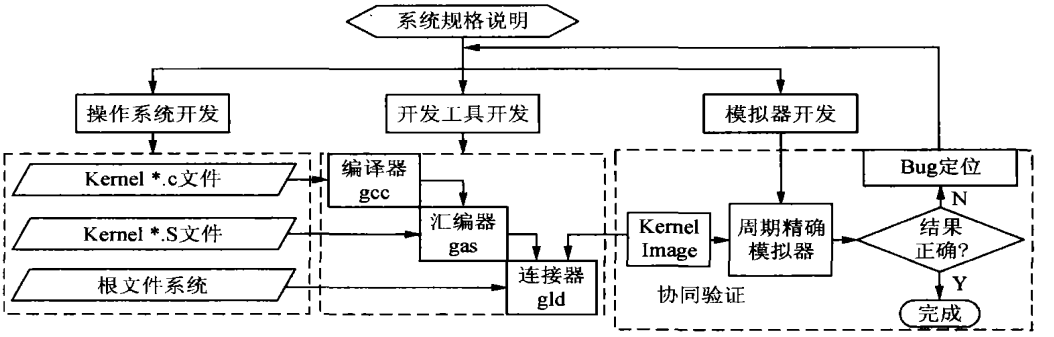

图 2 嵌入式 CPU协同验证过程示意图

5 结 论

本文提出一整套针对嵌入式CPU协同开发的操作系统设计方案,基于开放源代码软件,采用精简内核配置加内嵌RAMDISK的Linux Kernel,以uClibc和Busybox为主构成根文件系统,加入必要的基准测试程序,实现了精巧的嵌入式系统Tues-Linux。

在真正的硬件完成之前,通过利用虚拟原型系统进行软硬件集成测试和协同验证,基本完成一个完整的可运行Linux系统原型,并根据系统的应用领域,主要从设备驱动、系统工具、用户管理和日志、网络支持、网络服务以及图形用户界面等几个方面进行完善。

本方案成功应用于THUMP及其配套的系统软件和应用程序开发环境研发中,为NG网络流量分配器、网关等各种应用提供了稳定的系统支持,高效地完成了THUMP的协同验证,实现了最高主频达500MHz,功耗小于0.5W的THUMP-107这款CPU于2004年2月发布,是当时国内主频最高的CPU。

由此可见,本设计方案满足嵌入式CPU软硬件协同开发中验证目的和虚拟环境对操作系统设计提出的严格要求,同时也满足目标CPU实际运行的需求。目前,围绕这一方案的工作主要是实时性和低功耗改进。

致谢 感谢清华大学高性能所郑纬民、CPU中心汪东升两位教授提出的富有建设性的意见。感谢OS组和开发环境组的同学,本文所讨论的是大家共同的成果。

参考文献 (References)

[1] Wolf H W. A decade of hardware/software codesigh [J]. *Computer*, 2003, **36**(4): 38-43.

[2] Synopsys. Hardware/Software Co-design. [OL]. http://www.synopsys.com/products/hwsw/hwsw.html, 2003.

[3] Semeria L, Ghosh A. Methodology for hardware/software co-verification in C/C++ [A]. Proceedings of ASP-DAC 2000, Asia and South Pacific Design Automation Conference 2000 [C]. Yokohama, Japan: ACM, 2000. 405-408.

[4] Jain P P. Cost-effective co-verification using RTL-accurate C models [A]. Proceedings of IEEE International Symposium on Circuits and Systems 1999 [C]. Orlando, US: IEEE, 1999. 460-463.

[5] Farquhar E, Bunce P. The MIPS Programmer's Handbook [M]. San Fransisco: Morgan Kaufmann, 1994.

[6] Linux Kernel Organization. What is Linux [OL]. http://www.kernel.org, 2004.

[7] uClibc Group. A C Library for Embedded Linux. [OL]. http://www.uclibc.org/, 2004.

[8] Busybox Group. BusyBox: The Swiss Army Knife of Embedded Linux [OL]. http://busybox.net/, 2004.